ISO TC184/SC4/WG11 N123

Date: 2000-06-29 **Supersedes N072**

PRODUCT DATA REPRESENTATION AND EXCHANGE

Part: Title: EXPRESS-X Language Issue Log

Purpose of this document as it relates to the target document is:

	Primary content	Current status:	Working Draft
x	Issue discussion	Version	
	Alternate proposal		
	Partial content		

ABSTRACT:

Issue log against WG11 N066.

KEYWORDS

EXPRESS
EXPRESS-X
Mapping language

Document status/dates (dd/mm/yy)

Part Documents		Other SC4 Documents	
_____	Working draft	_____	Working
_____	Project draft	_____	Released
_____	Released draft	_____	Confirmed
_____	Technically complete	_____	Released
_____	Editorially complete		
_____	ISO Committee Draft		

Project leader:	Martin Hardwick	**Editor:**	Peter Denno
Address:	STEP Tools, Inc. 216 River Street Troy, NY 12180 USA	**Address:**	NIST 100 Bureau Drive Gaithersburg, MD 20899 USA
Telephone:	+1 (518) 687-2848	**Telephone:**	+1 (301) 975-3595
Fax:	+1 (518) 687-4420	**Fax:**	+1 (301) 975-4694
E-mail:	hardwick@steptools.com	**E-mail:**	peter.denno@nist.gov

Comments to Reader:

This document contains issues logged against the current EXPRESS-X language reference manual, document WG11 N066 and N110.

Issue 001: Make EXPRESS-X Inheritance same as EXPRESS
Classification: conceptual
Status: CLOSED

Date: 16 Dec 1998
Author: Martin Hardwick
Document clauses: Unknown

Problem Description:

The inheritance model for VIEWs in EXPRESS-X is different to the model used in EXPRESS. Specifically, the current syntax does not allow AND/OR inheritance. There does not appear to be any technical reason why EXPRESS-X VIEW inheritance is different to EXPRESS inheritance. Making e two different will confuse EXPRESS-X users who know EXPRESS.

Proposed Solution:

Allow EXPRESS-X to have the same inheritance model as EXPRESS by allowing a VIEW to use the same syntax as EXPRESS to declare its inheritance relationships

```
VIEW xyz
SUBTYPE OF (abc);
WHERE...

VIEW ghi ABSTRACT SUPERTYPE OF (ONEOF (jkl, mno));
WHERE...

VIEW pqr SUPERTYPE OF (mno, rst);
WHERE...
```

Date: 25 Jan 1999
Discussion at San Francisco meeting

MH: SUPERTYPE clause is a "post condition" that is checked after instantiation.

JV: IS it the case that a single binding is instantiated once for all views in the hierarchy that have a satisfied where clause?
MH: Yes.

JV: Is multiple inheritance allowed? MH: No.

MH takes action item to write this up and submit to editor.
Mapping of how VIEWs correspond to ENTITYs, so that we can discuss "the result of a view being an instance".
Will also write the description of "AND/OR inheritance" and what it means for VIEWs.

Issue 002: VIEW may be overloaded
Classification: conceptual
Status: CLOSED

Date: 16 Dec 1998
Author: Martin Hardwick
Document Clauses: Unknown

Problem Description:

Some VIEW definitions are used to create Virtual Entities
and some VIEW definitions are used to compute a value
from a population of entities. EXPRESS-X schemas may
be easier to understand if the two kinds of VIEWS used
different keywords.

Proposed Solution:

Use the FIND keyword for VIEWs whose purpose is to return a value.

```
FIND  result
FROM  (a:e1, b:v1)
WHERE a.x :=: b.y
SELECT
      a.z + 10;
```

Date: 17 Dec 1998
From: Steve Waterbury <steve.waterbury@gsfc.nasa.gov>

Martin Hardwick wrote:

> Proposed Solution:
>
> Use the FIND keyword for VIEWs whose purpose is to return a value.
>
> FIND   result
> FROM   (a:e1, b:v1)
> WHERE   a.x :=: b.y
> SELECT
>   a.z + 10;

"FIND" sounds like searching for something ... how about
"COMPUTE"?

Date: 25 Jan 1999
Discussion at San Francisco meeting

Peter to take action to write this up.
Replace SELECT with RETURN.

Issue 003: Syntax of FROM is an exception

Classification: syntax
Status: CLOSED

Date: 16 Dec 1998
Author: Martin Hardwick
Document Clauses: unknown

Problem Description:

The FROM clause in the header of a VIEW uses a parentheses driven syntax
while the other clauses are all ; driven. There does not seem to
be any justification for this difference.

Proposed Solution

Change the definition of the FROM clause to be ; like most
of the other keywords of EXPRESS.

```
VIEW xyz
FROM a:abc; b:def; g:hjk;
WHERE...
```

Date: 23 Jan 1999
From: John Valois

I propose that the FROM clause use the same syntax as function and
procedure parameters in EXPRESS, i.e.:

```
VIEW v;
FROM (a : atype; b, c : bctype)
```

[Parentheses are optional.]

Date: 27 Jan 1999
Discussion at San Francisco meeting

Remove parentheses; otherwise exactly like EXPRESS subroutine syntax.

```
VIEW v;
FROM a : atype; b, c : bctype;
 ...
```

Action taken by Gregor Lorenz to do this.

Issue 004: layout of view partitionsAutohr:
Classification: clarification
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document clauses: 9.3.1. view partitions

Problem Description:

Clause 9.3.1. states that all partitions must declare the same attributes including names and types.

I have the following problems:

What exactly is ment by the 'same' type?
I see to possible interpretations.
(a) SAME equals 'type compatibility'
(b) SAME equals 'EXACTLY the same type'

Proposed Solution:

If we agree to chose alternative (b), which I think is the correct interpretation, then I would like to propose NOT to specify an attributes type in each of the assignments. Instead I think it is reasonable to have a preluding attribute defininition section which has the purpose to specify all of the views attributes as well as their types.

EXAMPLE:

you would write

```
VIEW A;

   attribute1 : SET OF INTEGER;
   attribute2 : PERSON;

 PARTITION one: FROM source1, source2
   attribute1 := ...;
   attribute2 := ...;
 PARTITION two: FROM source3, source4
   attribute1 := ...;
   attribute2 := ...;
 END_VIEW;
```

instead of

```
VIEW A;
 PARTITION one: FROM source1, source2
   attribute1 : SET OF INTEGER := ...;
   attribute2 : PERSON = ...;
 PARTITION two: FROM source3, source4
   attribute1 : SET OF INTEGER := ...;
   attribute2 : PERSON = ...;
END_VIEW;
```

The alternate syntax proposal yields to an improved readability and a better (since less error prone and shorter) structure of the view declaration.

From: "Ian Bailey" <ian.bailey@eurostep.com>
Date: 8 Jan 1999

I like this proposal, but we will have to be careful with it. It must only be allowed when partitions are used, otherwise users will be able to define view attributes which are not derived from underlying data or set as constants. It would be meaningless to have unset (or underived) attributes in a view - it would be just like an entity.

I suspect there is a case for a third possibility - allowing the type (but not the name) of an attribute to be different between partitions. Surely this is the views equivalent of a SELECT type ?

```
VIEW A;
 PARTITION one: FROM source1, source2
  attribute1 : SET OF INTEGER := ...;
  attribute2 : PERSON = ...;
 PARTITION two: FROM source3, source4
  attribute1 : SET OF STRING := ...;
  attribute2 : ANIMAL = ...;
END_VIEW;
```

Personally, I hate SELECT in EXPRESS as it seems to cause many more problems than it solves, so I'm not going to push this idea too much !

Date: 25 Jan 1999
Discussion at San Francisco meeting

The types must be exactly the same.
No change to the syntax.

Issue 005: optional view attributes
Classification: conceptual
Status: OPEN

Date: 08 Jan 1999
Author: Gregor Lorenz
Document clauses: 9.3.1. view partitions

Problem Description:

How do we address the problem of OPTIONAL view attributes, i.e. attributes which need not to have a value?

(a1) Do we force the user to write an attribute assignment expression as for example "a := ?" ?

EXAMPLE:

```
VIEW person;
   first_name     : string;
   middle_initial : string;
   last_name      : string;
 PARTITION one; FROM source1;
   first_name := ...;
   middle_initial = ?;
   last_name  := ...;
 PARTITION two; FROM source2;
   first_name := ...;
   middle_initial = some_expression;
   last_name  := ...;
END_VIEW;
```

(a2) Do we allow the omittance of the attribute assignment statement?

EXAMPLE: In the above example partition 1 would be specified as

```
 PARTITION one; FROM source1;
   first_name := ...;
   last_name  := ...;
```

(b) I think there is currently no notion to declare some of the attributes to be either mandadory or optional.

Proposed Solution:

OPTIONAL has been added to the LRM

From: "Ian Bailey" <ian.bailey@eurostep.com>
Date: 8 Jan 1999

I might have misunderstood your issue...but here's my opinion on this:

Views do not have target schemas. The schema is implicit in the definition of the view. For this reason, attributes are only optional if there evaluation can be null - i.e if:

1) They are based on an optional attribute in the base (source) schema
2) They are evaluated from an explicit binding or in-line view where the number of returned instances (or values) may be zero.

In both these cases, it is possible for a software system to decide whether view attributes can be optional. The question is whether we allow users to use the OPTIONAL keyword when it is not strictly necessary. I guess Martin would support you on this, as he is pretty keen to replicate all the EXPRESS entity capabilities in views.

I like the idea of using the OPTIONAL keyword myself, but it is meaningless to define a view attribute as OPTIONAL if it plainly derived from a base attribute which is not optional. This means we would have to strictly define the circumstances under which the OPTIONAL keyword could be used.

The second point raises a fairly contentious issue that has been discussed many times in EXPRESS-X meetings. Inline views and partial explicit bindings can return between zero and unknown numbers of instances and values. The question has been raised many times whether we need to clarify this by putting an aggregate in the attribute definition - e.g.

```
view_attr : SET[0:?] OF base_entity := VIEW FROM base_entity WHERE ....
```

N066 does not use this in its inline view and partial explicit binding examples. I'm not clear on what the final decision on this was (it was discussed again at Beijing).

Can anyone else remember what was decided on this ?

Date: 23 Jan 1999
From: John Valois

Suggestion: Place no restrictions on the attributes declared in a partitions. Conceptually, think of mapping a view with partitions to a collection of EXPRESS entities like this:

```
VIEW v;
PARTITION one:
 ...
PARTITION two:
 ...
PARTITION three:
 ...
END_VIEW;

ENTITY v ABSTRACT SUPERTYPE;
  [no attributes]
END_ENTITY;

ENTITY one SUBTYPE OF (v);
  [attributes defined in partition one]
END_ENTITY;

ENTITY two SUBTYPE OF (v);
  [attributes defined in partition two]
END_ENTITY;

etc.
```

Date: 08 Jan 1999
From: Martin Hardwick <hardwick@steptools.com>

```
>view_attr : SET[0:?] OF base_entity := VIEW FROM base_entity WHERE ....
```

My hope is that a view attr should be defined as you have shown.
At one time examples like this were a big concern to me because I

was assumming that these examples would be occurring all
the time and converting results from being sets to singletons
would be a major pain. I was even going to suggest that we add
a new function such as SINGLE to make it easy to do the conversion

```
view_attr : base_entity := SINGLE (VIEW FROM base_entity WHERE ....)
```

However, one of the reasons I started to really like the INDENTIFIED_BY
function is that it in most cases it does the conversion. As you may
vividly remember we used to have horrible arguments about IDENTIFIED_BY.
The following convoluted example illustrates

```
view_attr : base_entity := something (argument);

VIEW something:base_entity;         -- Not current syntax (needs issue)!
FROM a:something;
WHERE condition;
IDENTIFIED_BY a.attribute_containing_argument;
SELECT
      a.attribute_containing_base_entity
END_VIEW;
```

The reason why this works is that without the IDENTIFIED_BY clause
the example has to be coded with an open parameter and additional
WHERE clause as in

```
view_attr : SET OF base_entity := something (argument, ?);

VIEW something;
FROM a:base_entity, b:argument_type;
WHERE a.attribute :=: b;
      condition;
```

In the above the system has to assume that the something VIEW
may return a set.

This is the end of my convoluted example and while I think I
have shown that IDENTIFIED_BY is a solution for explict views
I have not shown a solution for in-line views. There may be
an issue here because as far as I know there is no way to
specify IDENTIFIED_BY in an IN-LINE view. This may be another
issue

To change the subject, there have also been some messages on
OPTIONAL. I think the OPTIONAL keyword should be allowed for
an attribute because even with IDENTIFIED_BY it may be possible
that the VIEW will not return a value.

```
view_attr : OPTIONAL base_entity := something (argument);

VIEW something:base_entity;         -- Not current syntax (needs issue)!
FROM a:something;
WHERE condition;
IDENTIFIED_BY a.attribute_containing_argument;
```

```
SELECT
      a.attribute_containing_base_entity
END_VIEW;
```

I this example it is legal for an instance of view_attr to
contain a NULL value.

At the momement this only works for explicit views and cannot be used
for in-line views which may or may not be a weakness.

Date: 25 Jan 1999
Discussion at San Francisco meeting

MH: wants to have optional attributes. This is interpreted as a
post-condition check on the result of the expression. This also
requires placing restrictions on the expression on the rhs of
a view attribute definition; the expression can evaluate to '?'
only if the attribute is optional.

Issue 006: SUBTYPE OF in views is misleading
Classification: syntax
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.3.3. view inheritance

Problem Description:

> The keywords SUBTYPE OF are misleading, since a view is not a type.
> I know that this is basically a religious thing to start this
> discussion, but I believe VIEWS and ENTITIES are different concepts
> and we have to clarify what type compatibility and assignment
> compatitbility means if you can use views and entities
> interchangeably.

Proposed Solution:

> As for the keyoword issue, I would prefer SUBVIEW OF.
> The rest is to be discussed at San Francisco.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Issue is to change SUBTYPE keyword to SUBVIEW for views.
Dropped; keep SUBTYPE.

Issue 007: SUBTYPE OF clause in view declarations
Classification: clarification
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.3.3. view inheritance

Problem Description:

(a) The syntax allows referal to entity types in the SUBTYPE OF clause. The clause does not state the semantics of such a referal.

(b) What is the semantics if additional partitions are specified within a SUBVIEW ? Is it allowed at all? I came across this issue since there is a clause stating the a view must define one or more partitions. I do not think this is true for SUBVIEWS.

Proposed Solution:

to be discussed and included in the document ...

Date: 26 Jan 1999
Discussion at San Francisco meeting

For (a), SUBTYPE OF must be a view reference.
Action taken by Peter Denno to fix this in the rules and restrictions of appropriate clause.

For (b), no additional partitions can be specified.
Action taken by Guenter Sauter and Gregor Lorenze to provide more text for clause 9.5.4, and possibly some changes to grammar involving SUBTYPE.

Issue 008: extent and binding paramters of a SUBVIEW
Classification: clarification
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.3.3. view inheritance

Problem Description:

The general description of this clause is very fuzzy.

(a) May a subview be extended so that its extent is a superset of the superview's extent? This might be the case if you specify additional partitions for example.

(b) Are the subviews binding parameters the same as the ones of the superview, or is the superview's extent the binding of the subview?

Proposed Solution:

   To be discussed or clarified by John

Date: 26 Jan 1999
Discussion at San Francisco meeting

For (a), the answer is no. Also see issue 007 re: new partitions
in subtypes.

For (b), yes.

Issue 009: mandatory/optional network IDs
Classification: clarification
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.4.1. header of the map declaration

Problem Description:

  This clause is not clear when a network id is required or may
  be omitted.

Proposed Solution:

  If map_decl_header refers to one target entity type only, then the
  specification of a network id is prohibited. Otherwise mandatory.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno: map name is first identifier after MAP,
and network id is optional if only one.

Issue 010: replace group by network
Classification: editorial
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.4.1. header of the map declaration

Problem Description:

  The text uses 'group' as a synoym for 'network'.

Proposed Solution:

Replace 'group' by 'network'.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to fix this.

Issue 011: map attribute
Classification: clarification, editorial
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.4.2. partitions within a map declaration

Problem Description:

The clause states that all partitions must define the same attributes (including names and types).

Map attriubtes do not declare types - they refer to entity attributes which are already typed.

See related clause for view attributes. What if instantiations of a map paramter (i.e. target entity type) are optional?

E.g. MAP target : LIST [0:?] OF ...

Proposed Solution:

To be discussed. At least we should remove the '... and types' part.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to remove the offending text.

Issue 012: missing description for clause 9.5.2.
Classification: editorial
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.5.2. Inheritance

Problem Description:

clause is missing

Proposed solution:

  add some text :-)

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Martin Hardwick to do this.

Issue 013: restrict type map relationships to 1:1
Classification: clarification
Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.9. type map declaration

Problem Description:

 Syntax: What's are the implications of n:m mappings for the structure
 of the assignment statements?? Syntactically, the right hand side of
 such an assignment may be any kind of expression - especially any
 expression referring to more than one of the sources and TARGETS.

Proposed Solution:

  I believe that EACH of the assignment statements should describe the
  relationship, i.e. the transformation rule, between exactly ONE source
  and exactly ONE target.
  This is neither reflected by syntax, nor is it addressed by the 'rules
  and restrictions' section.

  I suggest to only support 1:1 mappings to avoid
  silly specifications as outlined below.

```
  TYPE_MAP dmark FROM dollar, yen;
     dmark = dollar * yen;
  END_TYPE_MAP.
```

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to incorporate changes.

Issue 014: explanation of type maps is fuzzy
Classification: clarification
Status: WITHDRAWN

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.9. type map declaration

Problem Description:

  The Rules and restrictions section states:

  (b) No more than two expressions; if second is omitted then reverse
      mapping is implicit.
  There may be actually more than TWO expressions since we currently
  allow n:m mappings. How can you guarantee to derive the implicit
  reverse mapping? Other than that, an assignment is a statement and
  not an expression.

  (c) the two expressions shall be inverses of each other
  see (b)

  (d) No entity instances shall be mapped by the TYPE MAP. The base type
      shall not be an entity type.
  What is meant by 'the base type'???

Proposed Solution:

  to be discussed

Issue 015: description of type maps
Classification: editorial
Status: WITHDRAWN

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.9. type map declaration

Problem Description:

  'The mapping is applied whenever the source attribute type is type
   compatible with one of the first types and the target attribtute
   type is type compatible with one of the second types.'

  What types are refered to by 'first' and 'second' type??

Probosed Solution:

   open

Date: 27 Jan 1999
Discussion at San Francisco meeting

Dropped, no longer an issue.
Related to issue 013.

Issue 016: INDEXING in instantiation loops
Classification: clarification

Status: CLOSED

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 13.1. FOR statement (instantiation loop)

Problem Description:

   INDEXING clause is not present and consequently the clause lacks
   a corresponding description.

Proposed Solution:

   add syntax and explanation

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to do this.

Issue 017: forward declarations of map partitions
Classification: editorial (?)
Status: CLOSED (duplicate of issue 007)

Date: 08 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.4. map declaration

Problem Description:

  In Beijing you discussed a concept for forward declaration of
  map partitions, i.e. partitions which are just named in a map
  but which are declared in (one or more?) map inheriting from it.

  Other than that you concluded to have an attribute assignment
  block containing assingments common to all partitions.

  These points are not reflected by the document.

Proposed Solution:

  Add the forward declarations the syntax and add some explanation
  as well.

Issue 018: Ambiguous use of nonterminal 'reference_clause'
Classification: syntax
Status: CLOSED

Date: 07 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2

Problem description:
Rule 87 defines the nonterminal 'reference_clause'. It is already used in ISO 10303-11 A.2. Since A.2 of the EXPRESS-X spec makes heavy use of other nonterminals defined in ISO 10303-11, this is misleading.
(The other redefined nonterminal is 'syntax', but there we'd say there is no danger of misinterpretation)

Proposed solution:
Change label of the nonterminal, for example to 'reference_clause_extended'. Change rule 100 accordingly

---

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to do this.

Issue 019: Allow VIEW's to specify a type for their result
Classification: conceptual
Status: CLOSED

Date: 11 Jan 1999
Author: Martin Hardwick

Problem description:

Sometimes a view is used to select data from the extent of a base entity. In this case the result of the view should have the type of the base entity. The type can be deduced from the definition of the SELECT clause, but for checking and clarity reasons it would be nice if the information modeler also had the option to specify the type in the VIEW clause.

(Other constructs in EXPRESS such as functions also allow the type of a result to specified redundantly.)

Proposed solution:
Allow the type of a result to be defined after the name of the view and seperate each using a ":"

```
VIEW  example:product;
FROM  pr:product;
WHERE pr.id > 10;
      pr.id < 100;
SELECT
      pr;
END_VIEW;
```

Date: 25 Jan 1999

Discussion at San Francisco meeting

[See issue 002.] Modify RETURN to specify type and remove type from header. Peter to include with work on issue 002.

Changed our mind; issue dropped.

Issue 020: conditional creation of map/view instances
Classification: conceptual
Status: CLOSED

Date: 12 Jan 1999
Author: Gregor Lorenz
Document Clauses: ?

Problem Description:

The CREATE statement does not allow the creation of instances, dependent on some condition over source data, i.e. an instance should be created only iff a specific condition holds.

Proposed solution:

Extend CREATE statement with a preluding WHERE clause specifying such a constraint.

Example:

```
CREATE APPCNT INSTANCE_OF application_context
WHERE SIZEOF(item) <> 0;
   application := '';
END_CREATE;
```

Date: 26 Jan 1999
Discussion at San Francisco meeting

This similar to a global rule; 'item' is the name of an extent and is treated as an implicitly defined variable (the set of all instances in the extent).

Is this allowed in the current document? It should be somewhere in clause 12?

Change syntax to (we also change INSTANCE_OF to colon):

```
      CREATE name ':' type [ FOR '(' extent { ',' extent } ')' ]
      [ WHERE domain_rule { ';' domain_rule } ]
        ...
```

Action taken by Gregor Lorenz to write up and submit to editor.

Action taken by Peter Denno to write up rules and restrictions:

  - CREATE allowed only in SCHEMA_MAP
  - identifier after the colon must refer to an entity
  - must have all appropriate attributes, etc.

Issue 021: exception handling
Classification: conceptual
Status: CLOSED

Date: 12 Jan 1999
Author: Gregor Lorenz
Document Clauses: affects the whole thing

Problem Description:

 As we have recognized several times before, there is a strong
 need to include some way for the user to get some feedback from
 the express-x engine if something goes wrong. Examples include
 conflicting (resp. inconsistent) source data when using
 IDENTIFIED_BY, violation of constraints, a report of all sources
 which have not been mapped.

Proposed solution:

 I'm not going to propose anything here, I just wanted it to be
 included in the issue log, so that we can back to it at some point
 in time.

Date: 26 Jan 1999
Discussion at San Francisco meeting

We would need to add some text to clauses 4 and 5 to define that there
were going to be defined exceptions, what an implementation is required
to do, etc.

Action taken by Peter Denno and Gregor Lorenz to do this.

Decision not to define formal exceptions, instead just add informal rules
and restrictions where appropriate.

Issue 022: INLINE_FUNCTION
Classification: clarification
Status: CLOSED

Date: 12 Jan 1999
Author: Gregor Lorenz
Document clauses: 12.3.

Problem Description:

 First of all - inline functions do not need arguments since you
 cannot call such a function (how would you bind the parameters?).

The return type is not needed either, since it is implicitely
given through the type of the attribute the result of the inline
function is assigned to.

Proposed solution:

remove parameter list and return type from the syntax

Add "Each possible path of the 'function' must return a value
compatible to the left hand side of the assignment stmt
in which the 'function' constitutes the right hand side."

to rules and restrictions.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Not an issue; no longer in the document?
It is in the grammar, but not in the text.
Action taken by Gregor Lorenz to remove from grammar.

Issue 023: Harmonization of Map header and FROM clause
Classification: syntax
Status: CLOSED

Date: 12 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.4.1 Header of the map declaration

Problem Description:

When instances of the same source play different roles within
a map/view, you may specify this in the FROM clause as follows.

FROM role1 role2 : source

Within the header, you have to write the following:

MAP role1 : source, role2 : source AS ...

That is one cannot specify multiple parameters for the same
target type.

Proposed Solution:

adopt syntax of from clause in the header

Date: 27 Jan 1999
Discussion at San Francisco meeting

Harmonize with issue 003, use same syntax.

Action taken by Gregor Lorenz to do this.

Issue 024: complex_entity_spec
Classification: syntax
Status: CLOSED

Date: 12 Jan 1999
Author : Gregor Lorenz
Document Clauses: ?

Problem Description:

The specification of complex entity types is currently denoted as follows.

entity1 AND entity2 AND entity3 ...

I do not like this notion, since it looks a little like a logical expressions. EXPRESS denotes complexes (although this is just in the Annex and not reflected by syntax) as

entity1 & entity2 & entity3 ...

which I think everybody will understand without even looking into the spec.

Proposed Solution:

Change AND to '&'.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Use & in order to harmonize with EXPRESS edition 2.
Action taken by Gregor Lorenz to do this.

Issue 025: consecutive semicolons
Classification: syntax
Status: CLOSED

Date: 12 Jan 1999
Author : Gregor Lorenz

Document Clauses: ?

Problem Description:

The map_attribute_declaration syntax is terminated with a semicolon. Some of the possible right-hand sides of this assignment are also terminated in that way, thus forcing the user to sometimes writing to consecutive semicolons.

The affected map_attr_assgnmt_expr possibilities are:

```
  foreach_expr
  forloop_expr
  inline_function_decl
  map_case_expr
  map_cond_attr_expr
```

Proposed Solution:

Remove trailing semicolon from the clauses mentioned above.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to fix this.

Issue 026: Harmonization of SOURCE/TARGET/REFERENCE clause
Classification: syntax
Status: CLOSED

Date: 12 Jan 1999
Author: Gregor Lorenz
Document Clauses: 11.1, 11.3.1, 11.3.2

Problem Description:

Both clauses (SOURCE and REFERENCE) support references to some kind of sources and facilitate some renaming.

While the REFERENCE clauses only allows renaming of items within some source schema, the SOURCE clause only allows renaming of the schema names itself. The statement regarding the SOURCE clause is alsow true for the TARGET clause.

IMO the two differten types of clauses complement each other and may be harmonized into a single clause facilitating both the renaming of schemas as well as the renaming of some items defined therein.

Proposed Solution:

replace both, reference_clause and source_interface_spec by the below production

```
SOURCE
  [ schema_alias ':' ] schema_ref
  [ REFERENCE
       resource_or_rename { ',' resource_or_rename }
  ] ';'
```

```
 change target_interface_spec to

 TARGET
   [ schema_alias ':' ] schema_ref
   [ REFERENCE
        resource_or_rename { ',' resource_or_rename }
   ] ';'
```

Date: 26 Jan 1999
Discussion at San Francisco meeting

Leave REFERENCE as is.

Action taken by John Valois to write text to say that the identifier after REFERENCE can be an EXPRESS-X keyword.

Action taken by Gregor Lorenz to make above changes to SOURCE and TARGET grammar within schema maps.

Issue 027: double dot in view_reference
Classification: syntax
Status: CLOSED

Date: 12 Jan 1999
Author : Gregor Lorenz
Document Clauses: syntax definition

Problem Description:

```
 view_reference = view_ref | primary_extended '.' view_qualifier .
 view_qualifier = '.' view_ref .
```

 This causes the problem that a in the second alternative of view_reference,
 a view_ref will always have to preceeding dots as qualifiers.

Proposed Solution:

```
 change view_reference to

 view_reference = view_ref | primary_extended view_qualifier .
```

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to fix this.

Issue 028: execution order of attr assgnmt
Classification: clarification

Status: CLOSED

Date: 12 Jan 1999
Author: Guenter Sauter
Document Clauses: 9.3.2/9.4.2

Problem Description:
Unclear semantics of the execution of attribute assignments. It is not clearly specified in the document in which order attribute assignments (i.e. the statements after the SELECT keyword) are executed by some mapping engine.

Proposed solution:
Add the statement that "the execution order of attribute assignments is arbitrary" to the document.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to write this text and add to document. Also add that as a consequence, assignments that depend on one another are legal but may have unpredictable results.

Issue 029: switch expression = if + case expression
Classification: conceptual
Status: CLOSED

Date: 12 Jan 1999
Author: Guenter Sauter

Document clauses: 12.7, 12.8

Problem Description:
The harmonization of the conditional expression, currently specified by the IF-THEN-ELSE concept, and the CASE expression were decided at the Beijing meeting in order to avoid conflicts with existing IF-THEN-ELSE statements.

Proposed Solution:

```
switch_expression =
   switch_if_expression | switch_case_expression .
switch_if_expression =
   SWITCH boolean_expression THEN attr_assgnmt_expr
   { ELSIF boolean_expression THEN attr_assgnmt_expr }
   ELSE attr_assgnmt_expr .
switch_case_expression =
   SWITCH selector OF
   case_label { ',' case_label } THEN attr_assgnmt_expr
   { case_label { ',' case_label } THEN attr_assgnmt_expr }
   ELSE attr_assgnmt_expr .
```

attr_assgnmnt_expr does not currently exist but only the view_attr_..
and the map_attr_.. A harmonization of map_attr_assgnmnt_expr and
view_attr_assgnmnt_expr might be useful.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to discuss this with Phil Spiby.
Tentatively accepted pending outcome of this discussion.

Issue 030: Assignment of single elements in aggregate target attr
Classification: conceptual
Status: CLOSED

Date: 12 Jan 1999
Author: Hirel, Kockelke, Sauter

Document clauses: 9.3, 9.4, Annex A

Problem Description:
It is not specified explicitly (also implicitly assumed out of EXPRESS),
that an aggregate target attribute may be assigned element by element.
For example if 'Produktgruppe' is a LIST or ARRAY attribute, is the use
of the indexing operator allowed?

```
MAP ...
SELECT
  Produktgruppen[1] := ...;
. Produktgruppen[2] := ...;
..
END_MAP;
```

or is it necessary to assign the target aggregate as a whole:

```
MAP ...
SELECT
Produktgruppen := ...;
...
END_MAP;
```

What about BAGs and SETs? Normally, the use of the indexing operator to
assign elements is not allowed (since these aggregates are not ordered).
Instead an operator '+=' may be introduced instead of having <set_attr>
:= <set_attr> + <new element> to append a new element to the aggregate.

Proposed Solution:
Attribute assignments of single elements are possible for target
attributes of type LIST and ARRAY (ie. .. SELECT attr[1] := expr;
attr[2] := expr; is allowed) as long as the same element is not assigned
more than once.
For target attributes of any type also the INLINE_VIEWs / FOR expression
are allowed to assign elements to a set. Furthermore the assignment of
sets to an attribute as defined in EXPRESS is allowed.

Date: 26 Jan 1999

Discussion at San Francisco meeting

We agree that the requirement is needed, but the syntax and semantics should be harmonized with EXPRESS TC2.

Not applicable with VIEW or with BAGs/SETs.

Action taken by Guenter Sauter to discuss with Phil Spiby.

Issue 031:
Classification: clarification
Status: CLOSED

Date: 12 Jan 1999
Author: Hirel, Kockelke, Guenter Sauter

Document clauses: 9.3, 9.4, Annex A

Problem Description:
It is not specified explicitly how to assign the attributes of each target instance to be created by a MAP in which the corresponding target entity is prefixed with 'LIST ... OF'. For example

```
MAP t1 : LIST [2:2] OF target_entity_1
...
SELECT
  t1[1].attr1 := ...;
...
  t1[1].attrn := ...;
  t1[2].attr1 := ...;
...
  t1[2].attrn := ...;
...
END_MAP;
```

Proposed Solution:
The assignment as described above is possible. Also, the FOR clause can be used to specify that many target instances shall be created for each qualified source instance. In that case, the FOR clause defines an iterator variable which can be used to index the target instances.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to write this up.

Issue 032: syntax + terminology + explanation of map_decl_header
Classification: clarification, syntax
Status: CLOSED

Date: 12 Jan 1999
Author : Gregor Lorenz

Document Clauses: 9.4.1

Problem Description:

What is regarded as renaming does not meet the semantics properly.
I think the rename for a target entity type is like the definition
of an out-parameter. When looking at a map, each map_declaration
describes how a number of target instances are created from a given
binding of source instances. Thus the entity types listed in the
FROM clause(s) act like formal parameters of a routine (in-parameters)
while those of the map_decl_header act like those of a function
returning multiple values (out_parameters).
Since this issue deals with the map_decl_header, it is also
related to a formerly posted issue regarding clause 9.4.1.

Proposed Solution:

change

```
  map_decl_header = target_entity_ref_list [ NAMED network_id ].
  target_entity_ref_list = target_entity_ref_list_el
                           { ',' target_entity_ref_list_el }.
  target_entity_ref_list_el = [ target_entity_alias_id ':'
                                [ LIST bound_spec OF] ]
                              target_entity_reference.
```

to

```
  map_decl_header = out_parameter_decls [ NAMED network_id ].
  out_parameter_decls = out_parameter_decl { ',' out_parameter_decl }.
  out_parameter_decl  = [out_parameter_id  { out_parameter_id } ':']
                        [ LIST bound_spec OF] target_entity_reference.

  out_paramter_ref = out_parameter_id.
```

Thus target_entity_ref_list and target_entity_ref_list_el are
superfluous.

[I would also like to suggest the term 'in_parameter' as a replacement
for a parameter name in the from_clause.]

The spec must then state that if no out_parameter_id is specified,
then there is an implicit definition of such a parameter with the same
id as the referenced entity.

The spec is also unclear about the usage of a network name. It is
not said when such a network name is mandatory and when it is optional.
I propose to include the following statement for clarification:

"A network name is mandatory if there is more than one
 out_parameter_decl. It is prohibited if there is only one such
 production"

Please note that multiple out-parameters of the same type may now
be defined simultaneously which is also related to a formerly posted

issue.

The introduction of out_parameter_id also has an impact on explicit binding (i.e. map_call) and the left hand side of map attribute assignments.

```
replace
  map_call = entity_reference [ '@' network_or_partition_qualification ]
             '(' expression { ',' expression }')'.

by
  map_call = out_parameter_ref [ '@' network_or_partition_qualification
]
             '(' expression { ',' expression }')'.


replace
  map_attribute_declaration = [entity_reference '.' ] attribute_ref
                              ':=' map_attr_assgnmt_expr.
by
  map_attribute_declaration = [out_parameter_ref '.' ] attribute_ref
                              ':=' map_attr_assgnmt_expr.
```

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to write this up.

Issue 033: optional creation of target instances
Classification: conceptual
Status: WITHDRAWN

Date: 12 Jan 1999
Author: Hirel, Kockelke, Sauter

Document clauses: 9.3, 9.4

Problem Description:
If a given MAP has many target entity names in its header, systematically all of them are created each time the MAP is performed. Sometimes however, the creation of some of the target instances may be conditional, depending on any criteria in the source instance(s). Currently, there are two ways to solve this problem:
7 create a separate MAP for each of these optionally created target instances, and make the explicit binding to MAPs conditional (using an IF statement),
7 or make 'LIST [0:1 or ?] OF' precede the 'optional' target entity in the MAP header.
Both ways are not convenient, since either they oblige to perform a totally unnecessary splitting of the MAPs, or they misuse (perhaps) the LIST feature in the MAP headers.

Proposed Solution:
Introduce an IF CASE Statement in the MAP-Body, for example:

```
MAP t1, t2
FROM s1, s2, s3
...
SELECT
  IF EXISTS(s1.attr1) THEN
    t1.attr1 := ...;
    ...
    t1.attrn := ...;
  END_IF;
  ...
END_MAP;
```

If none of the attributes of one target entity is set, no instance of this target entity should be created,
It should be possible to nest IF and CASE statements just as in EXPRESS or in the IF and CASE expressions of EXPRESS-X.
To avoid any misuse of this new feature, it may be ensured, that at least one of the target entity is instantiated, otherwise a runtime error may be returned.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Issue withdrawn by G?nter Sauter.

Issue 034: behavior of FOR expression/statement
Classification: clarification, editorial
Status: CLOSED

Date: 12 Jan 1999
Author: Hirel, Kockelke, Sauter

Document clauses: 12.6, 13.1

Problem Description:
The FOR expression and the FOR statement may use either the repeat control mechanisms from EXPRESS (REPEAT WHILE/UNTIL), or a newly introduced feature called 'EACH').
There is currently no statement concerning the behavior of 'FOR EACH' if the scope of the loop is empty, i.e. if the extent defined either through 'IN' or 'FROM ... WHERE' is empty.

Proposed Solution:
Precise that in the case the loop extent is empty, the FOR loop will be performed zero times.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to make this change to the document.

Issue 035: Partition in the subtype MAP/VIEW
Classification: clarification
Status: WITHDRAWN (see resolution of issue 007)

Date: 12 Jan 1999
Author: Hirel, Kockelke, Sauter

Document clauses: 9.2, 9.3

Problem Description:
A MAP declaration that refers to another MAP in which a supertype of the target entity is mapped uses the SUBTYPE_OF clause. In such a case, no new FROM or IDENTIFIED_BY clause may be defined but implicitly reuses the ones of the supertype MAP.
Why is it only possible to mention one PARTITION from the supertype MAP in the subtype MAP? It would be helpful to treat all concerned PARTITIONS in the same subtype MAP instead of creating a new subtype MAP for each of them.
The syntax definition of VIEWs is not symmetrical with the one of MAPs. In VIEWs it is possible to combine one supertype PARTITION (part of the SUBTYPE_OF clause) with many other partitions (part of the 'view_decl')! What is the meaning of this combination? Subpartitions of the supertype partition?

Proposed Solution:
Harmonize the 'map_decl' and the 'view_decl' syntax and allow many supertype PARTITIONs to be described in the same subtype MAP.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Subsumed by issue 007.

Issue 036: allow indexing of attributes in the body of a map
Classification: clarification
Status: WITHDRAWN (duplicate of issue 030)

Date: 12 Jan 1999
Author : Gregor Lorenz
Document Clauses: syntax

Problem Description:
This problem is caused by target attributes with a declared type being an aggregation type. The current syntax does not allow the specification of index qualifiers in the context of a map_attribute_declaration.

```
MAP t
FROM s
```

```
  SELECT
  (* BAD SYNTAX: *) items[1] := item;
  (* GOOD, but ugly *) items := [item]; (* usage of aggregate initializer
*)
  END_MAP
```

Proposed Solution:

```
  change map_attribute_declaration to read to read

  map_attribute_declaration = [out_parameter_ref '.']
                              attribte_ref {index_qualifier}
                              ':=' map_attr_assgnmt_expr ';'.
```

Date: 26 Jan 1999
Discussion at San Francisco meeting

Duplicate of issue 030.

Issue 037: Ambiguous use of nonterminal 'reference_clause'
Classification: duplicate of issue 018?
Status: WITHDRAWN

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2
Problem description:
Rule 87 defines the nonterminal 'reference_clause'. It is already used in ISO 10303-11 A.2. Since A.2 of the EXPRESS-X spec makes heavy use of other nonterminals defined in ISO 10303-11, this is misleading.
(The other redefined nonterminal is 'syntax', but there we'd say there is no danger of misinterpretation)
Proposed solution:
Change label of the nonterminal, for example to 'reference_clause_extended'. Change rule 100 accordingly

---

Date: 26 Jan 1999
Discussion at San Francisco meeting

Duplicate of issue 018.

Issue 038: Clarify use of semicolon in CASE construct
Classification: syntax
Status: CLOSED

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2, 12.8
Problem description:

Rule 41 does not allow an additional ';' at the end of a
case_expr_action. All examples in the document, e.g. EXAMPLE 22 (12.8),
have a semicolon that terminates each CASE-alternative.
Same situation arises for rules 72 and 128.
The case_expr as defined in rule 40 (et al) thus makes a CASE look
different than in standard EXPRESS. However, the lack of a semicolon
also occurs for conditional assignment using IF.

Proposed solution:
Either disallow semicolon in the case_expr, in this case change the
examples.
Or enforce the use of a semicolon to separate case actions. In this
case, change rule 41 to read
"case_expr_action = case_label { ',' case_label } ':' expression ';' ."
Change rule 72 to read
"map_case_expr_action = case_label { ',' case_label } ':'
     map_attr_assgnmt_expr ';' ."
Change rule 128 to read
"view_case_expr_action = case_label { ',' case_label } ':'
     view_attr_assgnmt_expr ';' ."
For the second alternative, it should then be considered to change
conditional assignment etc. to enforce a semicolon, too.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Change to harmonize with EXPRESS 2.
Action taken by Guenter Sauter to do this.

Issue 039: Avoid doubling semicolons
Classification: syntax
Status: WITHDRAWN (duplicate of issue 025)

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2
Problem description:
Rule 69 defines a map_attribute_declaration to end with a semicolon.
However, many (but not all) of the possible right-hand side (rhs) expressions
of the assignment in this rule are also specifying semicolons to terminate
their productions. Therefore in some situations, the grammar forces
two semicolons, for example
"id := IF x THEN y END_IF;;"

Proposed solution:
Since the rhs part of the assignment can also be some conventional
expression (see rule 69/66) which does not have a semicolon, we
propose to keep the semicolon in rule 69, and instead remove all
terminating semicolons in rules 51, 54, 60, 71, and 73.

Issue 040: FROM clause parameters
Classification: syntax
Status: CLOSED

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2, 15.5
Problem description:
According to A.2 rules 55, 56, 57, a FROM clause can look
like
"FROM a, b, c".
Rule 56 will also allow something like
"FROM x: y, a : b".
However, rule 56 also allows something like
"FROM a b c : x, d e f : y".
It is not clear from 15.5 what the semantics of
this collection of identifiers on the left side of a colon would be

Proposed solution:
We propose to simplify rule 56 to read:
"from_parameter = [parameter_id ':'] extent_reference ."
thus disallowing multiple parameter ids on the lefthand side.
Alternatively, we suggest to add explanatory text to clause 15.5
about the nature of this concept (e.g. "FROM a b : x declares the type
of a and b to be x").

Date: 27 Jan 1999
Discussion at San Francisco meeting

Clarify prose and harmonize with issue 003 and 023.
Action taken by Gregor Lorenz to do this.

Issue 041: Explicit binding operator simplification or clarification
Classification: clarification
Status: CLOSED

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2, 12.1
Problem description:
Rule 70 defines map_call to be an entity_reference, followed by
an optional '@'+network/partition ID, followed by a comma-list
of expressions in parenthesis.
However, rule 47 allows an entity_reference to begin with a
primary_extended, which can be - according to rules 84 and 85, e.g.
another map_call (or other complicated stuff). Therefore, a
map_call might be defined as
"x := a@b(c).d@e(f)" etc. The question is whether this kind of
variety is actually intended, since the examples in 12.1ff seem to
use only a simple entity reference before the '@' or '(' token.

Proposed solution:
Clarify whether original construct is needed/intended, and if so,
provide an example.
If not, simplify rule 70 to read e.g.
"map_call = entity_ref ['@' network_or_partition_qualification]
            '(' expression { ',' expression } ')' . "
or to read
"map_call = entity_ref [entity_qualifier] ['@'
  network_or_partition_qualification] '(' expression { ',' expression } ')' .
"
or whatever seems to be sufficient.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Duplicate of issue 032? No, not really.
Problem is with grammar.

Action taken by Gregor Lorenz to fix this in the grammar.

Issue 042: qualifier concatenation
Classification: clarification
Status: CLOSED

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2
Problem description:
Rule 84 defines a primary_extended to be a qualifiable_factor_extended
followed by an arbitrary number of qualifiers (qualifiers are in the
form ".something"). The question is, whether this concatenation of an
arbitrary number of qualifiers is needed/desired. In addition, the
nonterminal 'primary_extended' seems to be used on right hand sides of
productions only if followed by a (context-specific) qualifier
(e.g. rules 38, 47, 124, 138).

Proposed solution:
Provide guidance about when concatenated qualifiers are required, or
simplify rule 84 to read
"primary_extended = qualifiable_factor_extended ."

Date: 26 Jan 1999
Discussion at San Francisco meeting

Not really an issue, since we just copied what is in EXPRESS.

Issue 043: Avoid doubling dots
Classification: duplicate of issue 027?
Status: WITHDRAWN

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2
Problem description:
Rule 138, in conjunction with rule 137, specifies that a view_qualifier
can only be used with two '.' ("a..b"). We assume this is a typo.

Proposed solution:
Change rule 138 to read
"view_reference = view_ref | primary_extended view_qualifier . "

Issue 044: Allow [] on the left hand side of a map_attribute_declaration
Classification: conceptual
Status: WITHDRAWN

Date: 12 Jan 1999
Author: Helmut Kockelke/Jens Kuebler
Document clause(s) affected by the issue: A.2
Problem description:
Rule 69 does not allow an index_qualifier on the left hand side
of a map_attribute_declaration. This means that it is not possible
to assign to specific components of aggregate-type attributes of the
target entity, or at least it is not elegant.
For instance the following example is not allowed:
Source
```
------
Entity s;
   item: STRING;
END_ENTITY;
```
Target
```
------
Entity t;
   items: SET OF STRING;
END_ENTITY;

MAP t
FROM s
SELECT
(* WRONG: *) items[1] := item;
(* RIGHT, but ugly: *) items := (item);
END_MAP;
```

Proposed solution:
Allow an index qualifier on the left hand side, e.g. by changing rule 69 to
read:
"map_attribute_declaration = [entity_reference '.'] attribute_ref
                  [index_qualifier] ':=' map_attr_assgnmt_expr ';' ."

Date: 26 Jan 1999
Discussion at San Francisco meeting

Withdrawn; duplicate of issue 030.

Issue 045: definition of binding
Classification: editorial
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 3.3.1

Problem Description:
Unclear definition of "binding" and in particular what is meant by "(..) to the requirements of (..)"

Proposed Solution:
Probably, the constraints as specified by the WHERE and IDENTIFIED_BY clause are meant here.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to come up with a better definition and discuss over exploder.

Issue 046: definition of binding extent
Classification: editorial
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 3.3.2

Problem Description:
Unclear definition of binding extent and in particular which source data entity extens / view extents are meant here.

Proposed Solution:
Precise the definition.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to come up with better definition and discuss over exploder.

Issue 047: extent data type

Classification: editorial
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 8.3

Problem Description:
"Extent data types are established explicitly by VIEW declaration and implicitly by source EXPRESS schema ENTITY declarations."
What about target EXPRESS schema ENTITY declarations as used in SCHEMA_MAPs?

Proposed Solution:
Add "(..) implicitly by source or target EXPRESS (..)"

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to correct this sentence in document.

Issue 048: Error in example 5
Classification: editorial
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 9.2

Problem Description:
"SCHEMA_MAP iges2step; REFERENCE FROM" is the syntax for SCHEMA_VIEW.

Proposed Solution:
"SOURCE" and "TARGET" has to be specified instead of "REFERENCE FROM".

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to fix.

Issue 049: extending subtype views
Classification: clarification
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 9.3.3

Problem Description:
Extending subtype views by additional attributes is restricted by the document.

Proposed Solution:
Allow to extend subtype views by additional attributes.

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to remove this sentence from document.

Issue 050: semantics of partitions
Classification: clarification
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 9.3, 9.3.1, 9.4.3

Problem Description:
Unclear semantics of partitions.

Proposed Solution:
Add the following sentences to 9.3.1:
"Partitions are needed when a single view / target entity is related in a specific way to source data for some instances and differently to source data for some other instances. For example, you may distinguish between different kinds of persons in your source (e.g. employees, students, etc.) where those different entities are not related via generalization/classification (i.e. subtype/supertype relationships) but have only a single representation in your view / target schema (say "persons"). In that case, you want to relate the view / target entity 'person' to 'employees' as well as 'students' etc. Obviously, it is in all that cases the same view data type / the same MAP header. All other clauses (FROM, ID_BY, FOR, SELECT) can be different. That's why those clauses are grouped together in so-called partitions."

Date: 26 Jan 1999
Discussion at San Francisco meeting

Leave as is. Require partitions to always have a name.
Name of partition must be unique within the scope of that particular view/map.

Action taken by Guenter Sauter to add this to document.

Issue 051: Semantics if no ID_BY clause is specified

Classification: clarification
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 5 (sect. after ex. 1), 9.5.3

Problem Description:
Unclear semantics when no IDENTIFIED_BY clause is specified.

Proposed Solution:
Precise the document by adding the following statements to 9.5.3
"If no IDENTIFIED_BY clause is specified then the IDENTIFIED_BY clause is implicitly given by building the cartesian product over the OIDs of all FROM-clause entities."

Date: 26 Jan 1999
Discussion at San Francisco meeting

Don’t use the term "OID"; use something else?
Action taken Guenter Sauter to add some text to the document.

Issue 052: Move 9.5.5 to 9.4.4
Classification: editorial
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 9.5.5

Problem Description:
The inheritance for MAPs is different from that for VIEWs.

Proposed Solution:
Move 9.5.5 to 9.4.4

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to do this.

Issue 053: move 13.1 to 9.4.5
Classification: editorial
Status: CLOSED

Date: 13 Jan 1999
Author: Guenter Sauter

Document clauses: 13.1

Problem Description:
The FOR clause is rather a clause available for MAP declarations rather than a statement.

Proposed Solution:
Move 13.1 to 9.4.5

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Guenter Sauter to do this.

Issue 054: view_call_argument production is not needed
Classification: syntax
Status: CLOSED

Date: 14 Jan 1999
Author : Gregor Lorenz
Document Clauses: syntax

Problem Description:
  as of the current syntax, a view_call_argument is either an
  expression or another view_call. Since the view_call can also
  be specified with the help of expression syntax (more specifically
  with primary_extended), mentioning view_call in view_call_argument
  is redundant. Thus view_call_argument may be replaced by expression.

Proposed Solution:

  change
    view_call = view_reference '(' view_call_argument {',' view_call_argument }
                               ')'.
  to
    view_call = view_reference '(' expression {',' expression } ')'.

  and remove
    view_call_argument from the syntax

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to fix.

Issue 055: allow complexes as SOURCEs for Maps and Views
Classification: conceptual
Status: CLOSED

Date: 14 Jan 1999
Author : Gregor Lorenz
Document Clauses: from clause description

Problem Description:
  Sometimes, it is desireable to restrict the sources of a Map/View to
  fulfill certain constraints on their type.

  Currently you may only specify such a constraint in the WHERE clauses
  of a Map/View (resp. their partitions).

```
  MAP target_person
  FROM p : source.person
  WHERE 'alien' IN TYPEOF(p) AND 'female' in TYPEOF(P) AND
further_constraint
  ...
  END_MAP;
```

  In my opinion, it perfectly makes sense to directly restrict the
  source data-set by naming the complex type which constitutes a source.

```
  MAP target_person
  FROM p : alien & female  (* <<<<<<<<<<<<! *)
  WHERE furher_constraint
  ...
  END_MAP;
```

Proposed Solution:

  change syntax of from clause to support complexes

Date: 26 Jan 1999
Discussion at San Francisco meeting

Should be discussed with Phil Spiby.

Need to specify what a legal complex type is; can we reference
Annex B of EXPRESS LRM?

Issue dropped.

Issue 056: EXTENT built-in function/operator needed
Classification: conceptual
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: map_body/view_body

Problem Description:
  In a view body one can refer to all items visible in the Map/View.
  Thus a WHERE clause may specify constraints on items which are not

listed in the FROM clause. One may wish for example to execute a mapping only iff a constraint on items other than the in-paramters holds.

```
MAP t
FROM s1 (* in-parameter *)
WHERE SIZEOF(entity_type_ref) > 15;
...
END_MAP;
```

The problem I see is that 'source_entity_type' refers to a type and not an aggregate as required by the SIZEOF operator.

Thus, we need a way to get the extent of an entity.

```
MAP t
FROM s1 (* in-parameter *)
WHERE SIZEOF(EXTENT(entity_type_ref)) > 15;
...
END_MAP;
```

Proposed Solution:

Introduce new built-in operation 'EXTENT' to get the extent of an entity data type.
The return type of the EXTENT operation is 'SET [0:?] OF parameter_type'

[NOTE: we do not need this for a view since partial explicit binding does this job!]

Date: 26 Jan 1999
Discussion at San Francisco meeting

Agree that it is a requirement to refer to the extent.

Rejected the solution of using extent name as an implicit variable.

Action taken by Peter Denno to remove text with regard to above.

Action taken by Gregor Lorenz to talk to Phil Spiby regarding harmonization with EXPRESS 2.

Date: 27 Jan 1999
Discussion at San Francisco meeting

EXPRESS 2 will be using EXTENT, but the argument will be of type "type".

Use built in function called EXTENT, returns SET OF GENERIC, argument is of type STRING like ROLESOF.

Action taken by Gregor Lorenz to write up and put in document.

Issue 057: definition of 'evaluation' is misleading
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 3.3.3 evaluation

Problem Description:
  The first sentence reads:
  'the application of a binding to a view or map'

  You do not apply data TO an operation. The inverse is true.

Proposed solution:

  replace by:

  'the application of a view/map to a specific binding'

Date: 27 Jan 1999
Discussion at San Francisco meeting

Remove as definitions and expand in the fundamental practices section.
Action taken by Peter Denno to revise document.

Issue 058: undetailed declaration of 'map'
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 3.3.5 map

Problem Description:

 3.3.3 reads
 '... data of one (or more) source entity types and data of one
  (or more) target entity types'

 Relationship view data type to target entity type is missing.

Proposed solution:

 Replace by:
 '... data of one (or more) source entity types or view data types
  and data of one (or more) target entity types'

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to harmonize with issue 057.

Issue 059: view extent contains unecessary explanation
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 3.3.7 view extent

Problem description:

 the last sentence does not DEFINE what a view extent is. It's
 rather explanatory and does not belong to the definition of
 view extent.

Proposed solution:

 move last sentence to some other place in the document

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to do something here; perhaps use
a note.

Issue 060: unclear definition of view instance
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 3.3.9 view instance

Problem Description:

 The clause states that view instances are named. I do not think
 this is true. Other than that it uses the term 'view entity type'
 which is not defined.

Proposed solution:

 replace definition by:

 '3.3.9 view instance: an instance of a view data type'

Date: 27 Jan 1999

Discussion at San Francisco meeting

Change "named" to "identifiable" and "entity" to "data".
Action taken by Peter Denno.

Issue 061: Rules and Restrictions of SCHEMA MAP
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Doucment Clauses: 9.2

Problem Descriptions:

The last three points in the 'rules and restrictions' section need not to be mentioned since using the language elements (interface_specification, entity_decl, type_decl) is already prevented by syntax.

Other than that, the section lacks the 'Rules and restrictions:' header.

Proposed solution:
add header,
Remove these points from R&R.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to do both (add "Rules and restrictions", remove last three items).

Issue 062: missing detail in FROM clause description
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.3.1

Problem Description:

The first sentence reads: '... (view extents and entity extents) ...'

IMO we need to clarify that neiter of the above extents is mandatory in the FROM clause.

Proposed Solution:

replace by: ' ... (view extents and/or entity extents) ... '

Date: 27 Jan 1999
Discussion at San Francisco meeting

Change "and" to "or".
Action taken by Gregor Lorenz.

Issue 063: bad FROM clause in example 7
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.3.3

Problem Description:

  You cannot define an ENTITY 'number' since it is a built-in
  type of EXPRESS.

Proposed Solution:

  Replace 'number' by 'my_number'

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to fix.

Issue 064: R&R of IDENTIFIED_BY usage incomplete
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.3.3

Problem Description:

 "- view instances ... over source attributes values/OIDs."
 This section states that the IDENTIFIED_BY clause may only use
 expressions which use sources.

 This is not reflected by R&R.

Proposed Solution:

 Add
 'An expression in an IDENTIFIED_BY clause must not refer to
  the targets of the map or any of their attributes'

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to attempt to clarify.

Issue 065: usage of the term 'AGGREGATE'
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: the whole document

Problem Description:

The term AGGREGATE is used throughout the whole document in a misleading way in that it is used in a number of places where 'aggregation data type' is ment.

Proposed Solution:

change that!

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to fix this.

Issue 066: view attribute declaration unclear
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 9.4.1

Problem Description:

The text uses the terms 'role' and 'property' interchangeably. We should conlude on ONE terminology!

R&R (a) lists all possible types of map_attr_assgnmt_expr, which is not necessary.

R&R (b) is unclear about the indeterminate value '?'

Proposed Solution:

use either 'role' or 'property'

```
 replace R&R(a) by: "the view_attr_assgnmt_expr shall be ..."
 replace R&R(b) by: "every view attribute of a view instance shall
   have a value or indeterminate"

-------------------------------------------------------------------------
Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to fix this.

-------------------------------------------------------------------------
Issue 067: missing items in 'scope and indentifier defining items'
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 10,

Problem Description:

 There are at least schema _map and from_parameter_id missing
 as items declaring identifiers

Proposed Solution:

 Add items schema_map, from_parameter_id to Table2.
 Investigate what other items fit in this context

-------------------------------------------------------------------------
Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to add this to clause 10.

-------------------------------------------------------------------------
Issue 068: term 'view expression' not defined
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Auhtor: Gregor Lorenz
Document clauses: 10.3.2, 10.3.4

Problem Description:
 10.3.2 3rd paragraph and 10.3.4 use the term 'view expression'
 which is not defined.

Proposed Solution:

 use appropriate term or define 'view expression'

-------------------------------------------------------------------------
```

Date: 27 Jan 1999
Discussion at San Francisco meeting

Change to "view attribute", and define it.
Action taken by Peter Denno to do this.

Issue 069: scope, visibility of a view
Classification: clarification
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document clauses: 10.3.2

Problem Description:

10.3.2 states:

"visibility: ... is visible in the function, procedure, rule or schema view it is declared."

Views must not be declare in functions, procedures or rules. Furthermore, Views may be declared in SCHEMA_MAPs.

The clause also states:

"Scope: ... which terminates that entity declaration"

A view declaration is not an entity declaration!

Proposed Solution:

Replace
"visibility: ... scope of the SCHEMA_VIEW or SCHEMA_MAP in which.."

Replace
"Scope: ... which terminates the declaration."

"Scope:

Date: 26 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to make these changes, and also add similar clauses for SCHEMA_MAP.

Issue 070: I'm sick of inventing issue titles ...
Classification: editorial
Status: WITHDRAWN

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 11

Problem Description:

First paragraph, last sentence: "The REFERENCE specification..."
This belongs to 11.1

Last paragraph,
"The item referred to in the current schema by the new name if given
following the AS keyword."

This syntax (AS) only applies to the REFERENCE clause at the moment.
Nevertheless there is another issue proposing a harmonization of
TARGET, SOURCE and REFERENCE clauses. If we conclude on this
proposal the text remains okay.

Proposed Solution:

Move last sentence of first paragraph to 11.1

Replace the sentence in the last paragraph by:
"Furthermore, foreign items may be renamed to support the resolution
of name clashes, i.e. using the same name in more than one source,
and for renaming EXPRESS-X keywords used as item names in the
sources."

Issue 071: missing foreign sources
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 11.3, 11.3.1, 11.3.2, 11.3.3

Problem description:

11.3 states:
"A schema_map interface specifications allows items definec in
foreign schema to be visible withtin the schema map. ..."

Well, there is nothing said about referencing other SCHEMA_MAPs
and SCHEMA_VIEWs. This also applies to 11.3.3

11.3.1 and 11.3.2 uses singular instead of plural.

Proposed Solution:

replace 11.3, first sentence by:

"... defined in a foreign SCHEMA, SCHEMA_VIEW, or SCHEMA_MAP to ..."

in 11.3.1, 11.3.2 change 'schema' to 'schemas'

in 11.3.1 change to "... reference other SCHEMA_MAPs and/or
SCHEMA_VIEWs"

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to do this.

Issue 072: missing syntax, unresolved reference to figure
Classification: editorial
Status: WITHDRAWN

Date: 14 Jan 1999
Author: Gregor Lorenz
Document clauses: 11.6

Problem Description:
Syntax missing in the text, second paragraph contains an unresolved
reference reading "(cf.,???)"

Proposed SolutioN:
add missing syntax, resolve reference

Issue 073: SCHEMA_VIEW not mentioned in "importing mappings"
Classification: editorial
Status: CLOSED

Date: 14 Jan 1999
Author: Gregor Lorenz
Document Clauses: 15.3

Problem Description:

This clause does not mention references to SCHEMA_VIEWs.

Proposed solution:

"... in the referenced SCHEMA_MAP and/or SCHEMA_VIEW"

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to add appropriate text.

Issue 074: Execution Model Semantics not clear
Classification: conceptual, clarification
Status: CLOSED

Date: 15 Jan 1999
Author: Gregor Lorenz
Document clauses: Execution Model Semantics

Problem Description:
  I do not understand the description of the execution model semantics of
 EXPRESS-X. This is due to the fact that undefined terms (e.g. data
stream)
 are used, and that this clause speaks in terms of input and output but
does not
 describe what 'thing' operates on these streams, how they look like and
how
 data flows during runtime.

Proposed solution:
 This clause has been rewritten.

Issue 075: Matching explicit bindings to partitions
Classification: clarification
Status: WITHDRAWN

Date: 23 Jan 1999
Author: John Valois

Document clauses:

Problem Description:

The rules for matching an explicit binding to a partition need to be
clarified when the partitions of a view are over types that may be
compatible, e.g., subtypes of a common supertype.

Proposed solution:

To be discussed.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Not sure if this is really an issue; can the ambiguity be resolved
by using the concept that each binding must have a unique identity?
Action taken by John Valois to clarify and resubmit if an issue.

Issue 076: semicolon before SUBTYPE OF syntax
Classification: syntax
Status: CLOSED

Date: 23 Jan 1999
Author: John Valois

Document clauses:

Problem Description:

view_decl, subtype_of_clause:

To match EXPRESS, it should be:
      VIEW id SUBTYPE OF ... ;
not:
      VIEW id; SUBTYPE OF ...;

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Peter Denno to fix grammar to match EXPRESS.

Issue 077: entity_qualifier production unnecessary
Classification: syntax
Status: CLOSED

Date: 23 Jan 1999
Author: John Valois

Problem Description:

entity_qualifier, entity_reference:

Is entity_qualifier needed? Primary_extended can end in a qualifier.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by Gregor Lorenz to remove entity_qualifier from production 47.

Issue 078: VIEW syntax allows empty view
Classification: syntax
Status: CLOSED

Date: 23 Jan 1999
Author: John Valois

Document clauses:

Problem Description:

view_decl:

This allows an empty view; what does this mean:

      VIEW id; END_VIEW;

Date: 27 Jan 1999
Discussion at San Francisco meeting

Add a rules and restrictions to appropriate clause to require one
or the other, but not both, of FROM and SUBTYPE in VIEW declaration.

Action taken by John Valois to do this.

Issue 079: missing comma in from_parameter?
Classification: syntax
Status: WITHDRAWN

Date: 23 Jan 1999
Author: John Valois

Document clauses:

Problem Description:

from_parameter, from_parameter_list:

```
from_parameter =
  [ parameter_id { parameter_id } ':' ] extent_reference .
                ^^^
```

This is missing a comma?

```
from_parameter_list =
  from_parameter { ',' from_parameter } .
                   ^
```

This should be a semicolon, to match EXPRESS conventions.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Subsumed by issue 003.

Issue 080: syntax of SCHEMA.TYPE for view_attribute
Classification: clarification
Status: CLOSED

Date: 23 Jan 1999
Author: John Valois

Document clauses:

Problem Description:

Why are view_attributes of the form SCHEMA.TYPE necessary when
REFERENCE can be used to import the identifier?

Date: 27 Jan 1999
Discussion at San Francisco meeting

If there are a lot of name conflicts (likely if mapping) then this will
be much more convenient than renaming each symbol in the reference clause.

Issue 081: result of an invalid explicit or partial explicit binding
Classification: clarification
Status: CLOSED

Date: 23 Jan 1999
Author: John Valois

Document clauses:

Problem Description:

The result of a (partial) explicit binding that is invalid
should be indeterminate, in particular for a partial explicit
binding it should not be an empty set.

This allows writing constraints in the WHERE clause that check
for the existance of some structures as 'EXISTS view(x,y,?)'
rather than something like 'SIZEOF(view(x,y,?)) > 0'.

Date: 27 Jan 1999
Discussion at San Francisco meeting

Action taken by John Valois to write this up and submit to document
editor.

Issue 082: TYPE_MAP is a dangerous feature. It is unnecessary, since the
same functionality can be provided by functions.
Status: CLOSED

Date: 22 March, 2000
Author: Peter Denno

Document clauses:

Problem Description:

Date: 25 June, 2000
Discussion at Bordeaux meeting

We agree to remove TYPE_MAP from the specification.